Load the data

In [1]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
```

In [2]:
```
# Read the file into a variable fifa_data
fifa_data = pd.read_csv("fifa.csv")
```

In [3]:
```
# Showing the dataframe
fifa_data
```

Out[3]:

	Date	ARG	BRA	ESP	FRA	GER	ITA
0	1993-08-08	5.0	8.0	13.0	12.0	1.0	2.0
1	1993-09-23	12.0	1.0	14.0	7.0	5.0	2.0
2	1993-10-22	9.0	1.0	7.0	14.0	4.0	3.0
3	1993-11-19	9.0	4.0	7.0	15.0	3.0	1.0
4	1993-12-23	8.0	3.0	5.0	15.0	1.0	2.0
...	...	...	...	...	...	...	...
281	2018-02-15	4.0	2.0	6.0	9.0	1.0	14.0
282	2018-03-15	4.0	2.0	6.0	9.0	1.0	14.0
283	2018-04-12	5.0	2.0	8.0	7.0	1.0	20.0
284	2018-05-17	5.0	2.0	8.0	7.0	1.0	20.0
285	2018-06-07	5.0	2.0	10.0	7.0	1.0	19.0

286 rows × 7 columns

Examine & Analyse the data

In [4]:

```
# Print the 1st 5 rows of the data
fifa_data.head()
```

Out[4]:

	Date	ARG	BRA	ESP	FRA	GER	ITA
0	1993-08-08	5.0	8.0	13.0	12.0	1.0	2.0
1	1993-09-23	12.0	1.0	14.0	7.0	5.0	2.0
2	1993-10-22	9.0	1.0	7.0	14.0	4.0	3.0
3	1993-11-19	9.0	4.0	7.0	15.0	3.0	1.0
4	1993-12-23	8.0	3.0	5.0	15.0	1.0	2.0

In [5]:

```
# Print the last 5 rows of the data
fifa_data.tail()
```

Out[5]:

	Date	ARG	BRA	ESP	FRA	GER	ITA
281	2018-02-15	4.0	2.0	6.0	9.0	1.0	14.0
282	2018-03-15	4.0	2.0	6.0	9.0	1.0	14.0
283	2018-04-12	5.0	2.0	8.0	7.0	1.0	20.0
284	2018-05-17	5.0	2.0	8.0	7.0	1.0	20.0
285	2018-06-07	5.0	2.0	10.0	7.0	1.0	19.0

In [6]:

```
# Getting access to the shape of the data
fifa_data.shape
```

Out[6]:

```
(286, 7)
```

In [7]:

```
# Getting access to the index of the data
fifa_data.index
```

```
Out[7]: RangeIndex(start=0, stop=286, step=1)
```

```
In [8]: # Print the n rows of the data
        fifa_data.head(10)
```

Out[8]:

	Date	ARG	BRA	ESP	FRA	GER	ITA
0	1993-08-08	5.0	8.0	13.0	12.0	1.0	2.0
1	1993-09-23	12.0	1.0	14.0	7.0	5.0	2.0
2	1993-10-22	9.0	1.0	7.0	14.0	4.0	3.0
3	1993-11-19	9.0	4.0	7.0	15.0	3.0	1.0
4	1993-12-23	8.0	3.0	5.0	15.0	1.0	2.0
5	1994-02-15	9.0	2.0	6.0	14.0	1.0	7.0
6	1994-03-15	8.0	2.0	6.0	15.0	1.0	11.0
7	1994-04-19	10.0	1.0	7.0	15.0	2.0	13.0
8	1994-05-17	6.0	1.0	9.0	17.0	2.0	16.0
9	1994-06-14	8.0	3.0	5.0	13.0	1.0	4.0

```
In [9]: # Getting access to the columns of the data
        fifa_data.columns
```

```
Out[9]: Index(['Date', 'ARG', 'BRA', 'ESP', 'FRA', 'GER', 'ITA'], dtype='object')
```

```
In [10]: # Selecting one column
         fifa_data["Date"]
```

Out[10]:

```
0      1993-08-08
1      1993-09-23
2      1993-10-22
3      1993-11-19
4      1993-12-23
          ...
281    2018-02-15
282    2018-03-15
283    2018-04-12
284    2018-05-17
285    2018-06-07
Name: Date, Length: 286, dtype: object
```

In [11]:

```
# Selecting two or more columns
fifa_data[["Date","ARG",'ITA']]
```

Out[11]:

	Date	ARG	ITA
0	1993-08-08	5.0	2.0
1	1993-09-23	12.0	2.0
2	1993-10-22	9.0	3.0
3	1993-11-19	9.0	1.0
4	1993-12-23	8.0	2.0
...	...	...	...
281	2018-02-15	4.0	14.0
282	2018-03-15	4.0	14.0
283	2018-04-12	5.0	20.0
284	2018-05-17	5.0	20.0
285	2018-06-07	5.0	19.0

286 rows × 3 columns

In [12]:

```
# Obtaining the data type
type(fifa_data)
```

Out[12]:

```
pandas.core.frame.DataFrame
```

In [13]:

```
# Data type of each column
fifa_data.dtypes
```

Out[13]:

```
Date      object
ARG      float64
BRA      float64
ESP      float64
FRA      float64
GER      float64
ITA      float64
dtype: object
```

In [14]:

```
# Data type of that one column
type(fifa_data["Date"])
```

Out[14]:

```
pandas.core.series.Series
```

In [15]:

```
# Creating an array of 286 elements
new_col = np.arange(0,286)
```

In [16]:

```
# Creating random integer numbers between 1 to 100
new_col = np.random.randint(1, 100, size = 286)
```

In [17]:

```
# Creating random float numbers between 1 to 100
new_col = np.random.uniform(1, 100, size = 286)
```

In [18]:

```
# Adding a new column to a dataset with an array
fifa_data['new_col'] = new_col
```

In [19]:

```
# Showing the newly created data
fifa_data
```

Out[19]:

	Date	ARG	BRA	ESP	FRA	GER	ITA	new_col
0	1993-08-08	5.0	8.0	13.0	12.0	1.0	2.0	28.726307
1	1993-09-23	12.0	1.0	14.0	7.0	5.0	2.0	57.809709
2	1993-10-22	9.0	1.0	7.0	14.0	4.0	3.0	81.049730
3	1993-11-19	9.0	4.0	7.0	15.0	3.0	1.0	56.232458
4	1993-12-23	8.0	3.0	5.0	15.0	1.0	2.0	71.921954
...	...	...	...	...	...	...	...	...
281	2018-02-15	4.0	2.0	6.0	9.0	1.0	14.0	75.354410
282	2018-03-15	4.0	2.0	6.0	9.0	1.0	14.0	29.250439
283	2018-04-12	5.0	2.0	8.0	7.0	1.0	20.0	29.072796
284	2018-05-17	5.0	2.0	8.0	7.0	1.0	20.0	59.371469
285	2018-06-07	5.0	2.0	10.0	7.0	1.0	19.0	96.251986

286 rows × 8 columns

In [20]:

```
# Another methode of rounding numbers of new_col
fifa_data['new_col'].round(2)
```

Out[20]:

```
0      28.73
1      57.81
2      81.05
3      56.23
4      71.92
       ...
281    75.35
282    29.25
283    29.07
284    59.37
285    96.25
Name: new_col, Length: 286, dtype: float64
```

****

```
In [21]: # Showing the info of the dataset
         fifa_data.info
```

```
Out[21]: <bound method DataFrame.info of             Date   ARG  BRA   ESP    FRA  GER   ITA    new_col
         0    1993-08-08   5.0  8.0  13.0  12.0  1.0   2.0  28.726307
         1    1993-09-23  12.0  1.0  14.0   7.0  5.0   2.0  57.809709
         2    1993-10-22   9.0  1.0   7.0  14.0  4.0   3.0  81.049730
         3    1993-11-19   9.0  4.0   7.0  15.0  3.0   1.0  56.232458
         4    1993-12-23   8.0  3.0   5.0  15.0  1.0   2.0  71.921954
         ..          ...   ...  ...   ...   ...  ...   ...        ...
         281  2018-02-15   4.0  2.0   6.0   9.0  1.0  14.0  75.354410
         282  2018-03-15   4.0  2.0   6.0   9.0  1.0  14.0  29.250439
         283  2018-04-12   5.0  2.0   8.0   7.0  1.0  20.0  29.072796
         284  2018-05-17   5.0  2.0   8.0   7.0  1.0  20.0  59.371469
         285  2018-06-07   5.0  2.0  10.0   7.0  1.0  19.0  96.251986

         [286 rows x 8 columns]>
```

```
In [22]: # Select a column and find total sum
         fifa_data['ARG'].sum()
```

```
Out[22]: 1560.0
```

```
In [23]: # Select a row and find total sum
         fifa_data['ARG'] + fifa_data['ITA']
```

Out[23]:

```
0       7.0
1      14.0
2      12.0
3      10.0
4      10.0
       ...
281    18.0
282    18.0
283    25.0
284    25.0
285    24.0
Length: 286, dtype: float64
```

In [24]:

```
# Calculating the average score and assigning the result to a new column
fifa_data['Average'] = (fifa_data['ARG'] + fifa_data['ITA'])/2
fifa_data
```

Out[24]:

	Date	ARG	BRA	ESP	FRA	GER	ITA	new_col	Average
0	1993-08-08	5.0	8.0	13.0	12.0	1.0	2.0	28.726307	3.5
1	1993-09-23	12.0	1.0	14.0	7.0	5.0	2.0	57.809709	7.0
2	1993-10-22	9.0	1.0	7.0	14.0	4.0	3.0	81.049730	6.0
3	1993-11-19	9.0	4.0	7.0	15.0	3.0	1.0	56.232458	5.0
4	1993-12-23	8.0	3.0	5.0	15.0	1.0	2.0	71.921954	5.0
...	...	...	...	...	...	...	...	...	...
281	2018-02-15	4.0	2.0	6.0	9.0	1.0	14.0	75.354410	9.0
282	2018-03-15	4.0	2.0	6.0	9.0	1.0	14.0	29.250439	9.0
283	2018-04-12	5.0	2.0	8.0	7.0	1.0	20.0	29.072796	12.5
284	2018-05-17	5.0	2.0	8.0	7.0	1.0	20.0	59.371469	12.5
285	2018-06-07	5.0	2.0	10.0	7.0	1.0	19.0	96.251986	12.0

286 rows × 9 columns

In [25]:

```
# Describe the basic statistics of the dataset
fifa_data.describe()
```

Out[25]:

	ARG	BRA	ESP	FRA	GER	ITA	new_col	Average
count	286.000000	286.000000	286.000000	286.000000	286.000000	286.000000	286.000000	286.000000
mean	5.454545	3.171329	5.321678	8.958042	5.104895	8.353147	50.938657	6.903846
std	4.012659	3.620897	3.908787	6.822948	4.354813	4.514822	28.951252	2.810448
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.038914	1.500000
25%	3.000000	1.000000	2.000000	3.000000	2.000000	4.000000	27.443630	5.000000
50%	5.000000	1.500000	5.000000	7.000000	4.000000	8.000000	53.001715	6.500000
75%	7.000000	4.000000	8.000000	15.000000	5.000000	12.000000	76.482048	8.500000
max	24.000000	22.000000	25.000000	27.000000	22.000000	20.000000	99.579165	16.000000

In [26]:

```
# Rounding the values of the data set
round(fifa_data.describe(),2)
```

Out[26]:

	ARG	BRA	ESP	FRA	GER	ITA	new_col	Average
count	286.00	286.00	286.00	286.00	286.00	286.00	286.00	286.00
mean	5.45	3.17	5.32	8.96	5.10	8.35	50.94	6.90
std	4.01	3.62	3.91	6.82	4.35	4.51	28.95	2.81
min	1.00	1.00	1.00	1.00	1.00	1.00	1.04	1.50
25%	3.00	1.00	2.00	3.00	2.00	4.00	27.44	5.00
50%	5.00	1.50	5.00	7.00	4.00	8.00	53.00	6.50
75%	7.00	4.00	8.00	15.00	5.00	12.00	76.48	8.50
max	24.00	22.00	25.00	27.00	22.00	20.00	99.58	16.00

```
In [27]: # Length of the dataset [no. of rows]
         len(fifa_data)
```

```
Out[27]: 286
```

```
In [28]: # Highest index of the dataset
         max(fifa_data.index)
```

```
Out[28]: 285
```

```
In [29]: # Lowest index of the dataset
         min(fifa_data)
```

```
Out[29]: 'ARG'
```